PATENT ABSTRACTS OF JAPAN

(11)Publication number : **06-162133**

(43)Date of publication of application : **10.06.1994**

(51)Int.Cl. **G06F 15/60**

(21)Application number : **04-318180**

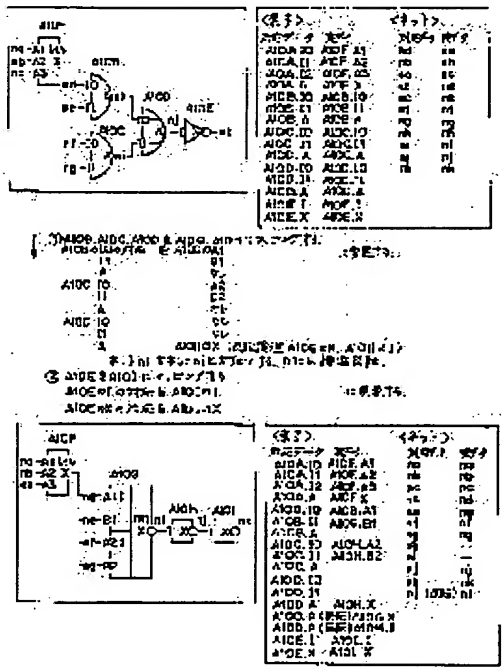
(22)Date of filing : **27.11.1992**

(71)Applicant : **FUJITSU LTD**

(72)Inventor : **FUKUSHIMA HISAYO**

(54) METHOD FOR SAVING CORRESPONDENCE RELATION OF FUNCTION DIAGRAM AND CIRCUIT DIAGRAM

(57)Abstract:



PURPOSE: To save the correspondence relations of the function diagram and circuit diagram more accurately by saving the correspondence relations between terminals of elements of the function diagram and a network and terminals of elements of the circuit diagram and a network including cases wherein logic inversion is caused between the terminals or networks.

CONSTITUTION: When AND macros A10B and A10C, one OR macro A10D, and an inverter macro A10E are mapped to a cell A10G and inverter cells A10H A10I, input terminals A1 and B1 of an element A10G of the circuit diagram correspond to input terminals 10 and 11 of an element A10B of the function diagram, input terminals A2 and B2 of the element A10G of the circuit diagram correspond to input terminals 10 and 11 of all element A10C of the function diagram, and output terminal X of the element A10H, the output terminal of the element A10G, and the input terminal of the A10H of the circuit diagram correspond to the output terminal A of the A10D of the function diagram by logic inversion; and those relations are saved. Further, a state wherein a network nj of the function diagram corresponds to a network n1 of the circuit diagram by logic inversion is saved.

LEGAL STATUS

[Date of request for examination] 03.07.1997

[Date of sending the examiner's decision of rejection]	
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	3002066
[Date of registration]	12.11.1999
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	12.11.2004

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application] This invention relates to the store method of the correspondence relation between the functional diagram which consists of components independent of technology, and the circuit diagram which consists of components depending on technology.

[0002]

[Description of the Prior Art] In the design of LSI, it is necessary to change into the logical circuit (for it to be hereafter called a circuit diagram) based on the component which depends on technology realizable on LSI for the logical circuit (it is hereafter called a functional diagram) designed based on the component independent of technology. In that case, in order to perform the logic simulation of the circuit after conversion etc., it is necessary to save the correspondence relation between a functional diagram and a circuit diagram.

[0003] Drawing 19 and drawing 20 are the explanatory views of the store method of the correspondence relation between the conventional functional diagram and a circuit diagram. In the conventional store method, the associated data in which the component (it is hereafter called a macro) which does not depend for the correspondence relation of each component on the technology of a functional diagram as shown in drawing 19 is shown, and the live data which show the component (it is hereafter called a cel) depending on the technology after mapping were made to correspond, and it saved. Moreover, the associated data of the network of a functional diagram and the live data of the network of the circuit diagram after mapping were made to correspond, and the correspondence relation of a network was saved.

[0004] Since the component of the circuit diagram corresponding to the component of a functional diagram does not exist yet in the phase where the functional diagram was created, it is drawing 19 (1). The component and network of a functional diagram are defined as live data corresponding to the component and network of a functional diagram so that it may be shown.

[0005] When AND macro A10A is mapped in AND cel A10F in this condition, it is this drawing (2). AND cel A10F are saved as live data corresponding to associated data A10A of a component so that it may be shown. In this case, since there is no modification of a network,

each network data is saved as it is.

[0006] Next, AND macro A10B, A10C, and A10D are mapped in cel A10G which consist of two AND and NOR, and inverter cel A10H, inverter macro A10E is mapped in inverter cel A10I, and it is drawing 20 (3). The shown circuit diagram is created.

[0007] At this time, A10G and A10H are saved as live data corresponding to associated data A10D which shows component A10D of a functional diagram as component data. Moreover, it is saved as a component with which A10I represents the original component as live data corresponding to associated data A10E which shows component A10E of a functional diagram, and the live data of component A10B and A10C are deleted.

[0008] Furthermore, although nl is newly defined as live data of a network, the associated data corresponding to the network nl of a circuit diagram will not exist. Moreover, networks nh and ni of a functional diagram Since it does not exist in a circuit diagram, the live data corresponding to those networks are deleted.

[0009] Drawing 20 (3) Since it is omissible, inverter cel A10H and A10I which exists in a serial with a circuit diagram is this drawing (4). As shown in a circuit diagram, inverter A10H and A10I are deleted.

[0010] At this time, A10E is deleted from the associated data of a component, and they are A10H and A10I from the live data of a component. It is deleted. Moreover, they are Networks nj and nk on a circuit diagram. Since it stops existing, they are live data nj and nk. It is deleted.

[0011]

[Problem(s) to be Solved by the Invention] In the store method correspondence-related [conventional] mentioned above, when two AND macro A10B and A10C and one OR macro A10D were mapped in cel A10G with two AND+NOR functions, and inverter cel A10H, for example, AND macro A10B of a functional diagram and the correspondence relation between A10C and the component of a circuit diagram were not saved. Moreover, when inverter cel A10H and A10I are deleted for optimization of a circuit, the correspondence relation between the inverter macro of a functional diagram and the component of a circuit diagram has disappeared.

[0012] since [namely,] only the component of the functional diagram before mapping, the component of the circuit diagram,after a network and mapping, and the correspondence

relation of a network are saved in the correspondence-related store method of the conventional functional diagram and circuit diagram which were mentioned above -- one pair ** -- many -- when pair 1 or many to many mapping is performed, the component of a functional diagram, the component of a network and a circuit diagram, and correspondence of a network can save correctly -- ** -- there was a trouble to say.

[0013] Moreover, when a component was deleted for optimization at the time of mapping of a circuit, there was a trouble that the correspondence relation between the component of a functional diagram and the component of a circuit diagram will disappear. Consequently, when performing the simulation of the circuit after mapping, and the terminal or network of a functional diagram may be unable to be specified as the trace point and simulation of the circuit after mapping was performed, there was un-arranging.

[0014] The purpose of this invention is enabling it to save the component of a functional diagram, the component of a network and a circuit diagram, and the correspondence relation of a network also at the time of one-pair ** and many to many mapping. Furthermore, also when a component is deleted for optimization of a circuit, it is enabling it to save the correspondence relation between a functional diagram and a circuit diagram.

[0015]

[Means for Solving the Problem] In the approach of saving the correspondence relation between the component independent of technology, the functional diagram constituted by the network, the component depending on technology, and the circuit diagram constituted by the network, it saves with the logic reversal information which shows the existence of the logic reversal of the correspondence relation between the input/output terminal of the component of the above-mentioned functional diagram, and the input/output terminal of the component of the above-mentioned circuit diagram between input/output terminals by the correspondence relation store method of the functional diagram and the circuit diagram of this invention.

[0016] Furthermore, it saves with the logic reversal information which shows the existence of the logic reversal of the correspondence relation between the network of the above-mentioned functional diagram, and the network of the above-mentioned circuit diagram during a network.

[0017]

[Function] Two individuals independent of the technology shown in drawing 12 [2 For

example, and macro A10B, A10C, one OR macro A10D, and inverter macro A10E This drawing (3) When it maps in cel A10G, inverter cel A10H, and A10I depending on the shown technology, in the correspondence relation store method of this invention The input terminals A1 and B1 of component A10G of a circuit diagram are equivalent to the input terminals I0 and I1 of component A10B of a functional diagram. The input terminal A2 of component A10G of a circuit diagram and B-2 correspond to the input terminals I0 and I1 of component A10C of a functional diagram. To the output terminal A of component A10D of a functional diagram The output terminal X of component A10H of a circuit diagram It is saved that the output terminal of component A10G and the input terminal of A10H correspond by logic reversal.

[0018] Furthermore, it is saved that the network nj of a functional diagram corresponds with the network nl of a circuit diagram by logic reversal. Thus, it becomes possible to specify the terminal and network of a component of the functional diagram which was not able to be conventionally specified as the trace point of a logic simulation as the trace point by saving the correspondence relation between the terminal of the component of a functional diagram, a network, and the terminal of the component of a circuit diagram and a network, and saving the information which shows the existence of the logic reversal between those terminals and a network further. Thereby, the logic simulation of gate level can be performed more correctly.

[0019] Furthermore, also when the component of a circuit diagram is deleted for optimization of a circuit, the correspondence relation between the input/output terminal of the component of a functional diagram and the input/output terminal of the component of the circuit diagram which corresponds by logic reversal is saved.

[0020] Therefore, on a circuit diagram, although it exists on a functional diagram, even if it is the component deleted, the input/output terminal of the component can be specified as the trace point, and simulation of the circuit after composition can be performed efficiently.

[0021]

[Example] Hereafter, the example of this invention is explained, referring to a drawing. Drawing 1 is drawing showing the functional diagram and circuit diagram which were created based on the correspondence relation store method of this invention, and the data then saved. In addition, the associated data of this drawing shows the terminal and network of a component of a functional diagram, and live data show the terminal and network of a

component of a circuit diagram.

[0022] In the phase which created the functional diagram, the terminal and network of a component of a functional diagram are saved as live data. Drawing 1 (1) If connection of 3 input AND macro A10A [which is shown], AND macro A10B, A10C, OR macro A10D, and inverter macro A10E is defined and a functional diagram is created As the associated data and live data of a component, A10A.I0, A10A.I1, A10A.I2, ..., etc. are saved, and na, nb, nc, etc. are saved as the associated data and live data of a network. In addition, this drawing (1) A10A.I0 of associated data and live data, A10A.I1, and A10A.I2 The input terminal of a 3 input AND macro is shown, A10A.A shows the output terminal, and other components are the same.

[0023] First, the configuration of the table which saves the component of a functional diagram and a circuit diagram and the correspondence relation of a network is explained with reference to drawing 2 . Drawing 2 is drawing showing the configuration of a managed table, the component table of live data, a terminal table, and a network table.

[0024] The data which point out the component table of the head of live data, the network table of the head of a network, the correspondence component table of the head of a correspondence component, and the correspondence network table of the head of a correspondence network as shown in drawing 2 are stored in a managed table.

[0025] The pointer which points out the terminal table of the head of an input terminal, the pointer which points out the terminal table of the head of an output terminal, the pointer which points out the terminal table of the head of an input/output terminal, the function of a component, a component proper name, and the pointer that points out the following component table are stored in a component table.

[0026] Moreover, the pointer which points out the component table of the component which belongs, the pointer which points out the network table of the network to connect, a terminal name, and IO classification are stored in a terminal table. Moreover, although not shown in this drawing, the pointer which points out the next terminal table of the same component to a terminal table is formed.

[0027] Furthermore, the pointer which points out the terminal table of a supplying agency, the pointer which points out the terminal table of the head of the terminal of a supply place, the pointer which points out the head terminal table of the terminal in two ways, the pointer

which points out a correspondence network, a network name, and the pointer which points out the following network table are stored in the network table of live data.

[0028] Here, the concrete contents of the managed table mentioned above with reference to drawing 3 and drawing 4 , a component table, a terminal table, and the network table are explained. In the phase where the functional diagram of drawing 1 was defined first, it considers as the head table of a component, RA10A considers [E1 / NA] as the head table of a correspondence network as a head table of a correspondence component as a head table of a network, and RNA is stored in the managed table.

[0029] The data in which it is shown that the head table of an input terminal is the terminal table P1 and that the head table of an output terminal is the terminal table P4 are stored in the component table E1 of AND macro A10A shown in drawing 3 . In this case, since the input/output terminal is not defined, no data which point out the head table of an input/output terminal are stored. Furthermore, the data in which it is shown that the function of this component is AND, that a component name is A10A, and that the following component table is E2 are stored in the component table E1.

[0030] The data in which it is shown that the following terminal table is P2, that the component table on which a terminal I0 belongs is E1, that the network to connect is NA, that a terminal name is I0, and that terminal classification is an input terminal I are stored in the terminal table P1 which is a head table of the input terminal of the component table E1. Hereafter, the same data also as each terminal tables P2 and P3 are stored.

[0031] Moreover, the data in which it is shown that the following output terminal table does not exist, that the component table on which an output terminal A belongs is E1, that the network table which an output terminal A connects is ND, that a terminal name is A, and that terminal classification is an output terminal O are stored in the terminal table P4 which is a head table of the output terminal of the component table E1.

[0032] The data which point out the supply place address table which stored the address of the terminal table of a supply place are stored in the pointer which points out the terminal table of the supply place of the network table NA of drawing 4 . And the address data in which the terminal table P1 is shown, and existing [the other terminal table] **** data are stored in the supply place address table.

[0033] Furthermore, the data in which it is shown that the terminal table of a supplying

agency is not defined in this case, that the terminal in two ways of a supply place does not exist, that a correspondence network is RNA, that a self network name is NA, and that the following network table is NB are stored in the network table NA.

[0034] The supply origin of Network na is not defined but these data show that a supply place is the terminal I0 of component A10A. Moreover, the pointer which points out a supplying agency address table, and the pointer which points out a supply place address table are stored in the pointer which points out the terminal table of the supply origin of the network table ND of drawing 4 . And the address data of the terminal table P4 and existing [the other terminal table] **** data are stored in the supply former address table. Moreover, the address data of the terminal table P5 and the data in which it is shown that the other terminal table does not exist are stored in the supply place address table.

[0035] The supply origin of Network nd is the output terminal A of component A10A, and these data show that a supply place is the input terminal I0 of component A10B. Next, the configuration of a correspondence component table, a correspondence terminal table, and a correspondence network table is explained with reference to drawing 5 .

[0036] The pointer which points out the head terminal table of a correspondence terminal, the function of the original component, the proper name of the original component, and the pointer that points out the following correspondence component table are stored in the correspondence component table.

[0037] Moreover, the pointer which points out the correspondence component table on which the terminal belongs, the pointer which points out a corresponding terminal table (live data), the pointer which points out the terminal table which corresponds by logic reversal, the original terminal name, and the pointer which points out the following terminal table are stored in the correspondence terminal table.

[0038] Furthermore, the pointer which points out the correspondence terminal table which a network connects, the pointer which points out a corresponding network table, the pointer which points out the network table which corresponds by logic reversal, and the pointer which points out the following network table are stored in the correspondence network table.

[0039] Next, the concrete contents of the correspondence component table mentioned above with reference to drawing 6 and drawing 7 , a correspondence terminal table, and the correspondence network table are explained. In the phase where the functional diagram of

drawing 1 was defined, the data in which it is shown that the head table of the terminal of the component is the correspondence terminal table RP 1, that the function of the original component is AND, that the original component name is A10A, and that the following component table is RA10B are stored in correspondence component table RA10A of drawing 6 .

[0040] The data which point out a specific address table are stored in the pointer which points out the terminal table of the correspondence terminal table RP 1, and the address data of terminal TEBURURU P1 corresponding to the correspondence terminal table RP 1 are stored in the address table.

[0041] Furthermore, the data in which it is shown that the correspondence component table on which the terminal belongs is RA10A, that the terminal table which corresponds by logic reversal does not exist, that the original terminal name is I0, and that the following correspondence terminal table is RP2 are stored in the correspondence terminal table RP 1.

[0042] These data show that the input terminal I0 of component A10A is equivalent to the input terminal I0 of component A10A of a functional diagram. In this case, since composition of a circuit diagram is not performed yet, the components of a functional diagram correspond by 1 to 1.

[0043] Moreover, the data in which it is shown that a corresponding network table is NA, that the network table which corresponds by logic reversal does not exist, and that the following correspondence network table is RNB are stored in the correspondence network table RNA of drawing 7 .

[0044] Moreover, the pointer which points out the correspondence terminal table of the correspondence network table RNA has pointed out the specific address table, and the address data of the terminal table RP 1 linked to the correspondence network table RNA are stored in the address table.

[0045] In this case, except the address of the correspondence terminal table RP 1, these data show that the network na of a functional diagram connects only with the input terminal I0 of component A10A at the above-mentioned address table without being stored.

[0046] Moreover, the address data of the correspondence terminal table RP 4 and the address data of the correspondence terminal table RP 5 are stored in the address table specified on the correspondence network table RND of drawing 7 .

[0047] These data show that the network nd of a functional diagram is connected to the output terminal A of component A10A, and the input terminal I0 of component A10B. Thus, when give the information on the terminal linked to each network to the network of a circuit diagram, and the network of a functional diagram, making the network of a functional diagram, and the network of a circuit diagram correspond further and replacing a component, correspondence relation between the network on a circuit diagram, modification of connection of a terminal, and the circuit diagram and functional diagram that were changed can be saved by changing the initial entry of a network, and the initial entry of a terminal at coincidence. Thereby, correspondence relation of the mapping, functional diagram, and circuit diagram of the component of a circuit diagram can be saved efficiently. Moreover, the terminal linked to a network and its network can be easily searched with using the table of structure which was mentioned above.

[0048] Next, it is drawing 1 (1) on the assumption that each table of the above configurations. About AND macro A10A of a functional diagram, it is this drawing (2). The actuation in the case of mapping in AND cel A10F is explained with reference to the flow chart of drawing 8 .

[0049] If AND macro 10A is mapped in AND cel A10F, the component table and input/output terminal table of cel A10F which should be replaced will be first created at step S1 of the flow chart of drawing 8 .

[0050] Drawing 9 is drawing showing the contents of the component table and terminal table which are created at this time, and the managed table changed. First, the component table E6, terminal table P16P, and P17, P18 and P19 are created, and the terminal table P16 is set up as a head table of the input terminal of the component table E6, and the terminal table P19 is set up as a head table of an output terminal. In this phase, since the input/output terminal of a component and connection of a network are not defined, the connection data to the network table of each terminal tables P16, P17, P18, and P19 are not set up.

[0051] Moreover, since it registers with the head table of the component of a managed table, component A10A replaced this time is changed so that the component table E6 of component A10F which newly created the pointer which points out the head table of a component may be pointed out.

[0052] If the component table and terminal table of a component which are newly mapped are created, it will be MAPP (A10A.I0, A10F.A1) of return and step S2 to drawing 8 . Processing

is performed. Drawing 10 is Above MAPP (A10A.I0, A10F.A1). It is the more detailed flow chart of processing.

[0053] First, it asks for the network table NA which the terminal connects from the terminal table P1 of the terminal I0 of A10A of live data at step S11 of drawing 10 . It asks for the correspondence network table RNA which corresponds from the above-mentioned network table NA at the following step S12.

[0054] Furthermore, at step S13, it asks for the correspondence terminal table RP 1 from the correspondence network table RNA, and the terminal table on which the correspondence terminal table RP 1 corresponds is changed into the terminal table P16 of the input terminal A1 of newly mapped component A10F from the terminal table P1 of the input terminal I0 of component A10A.

[0055] It asks for the network table NA linked to a terminal I0 from the terminal table P1 of component A10A of drawing 3 , and, specifically, asks for the correspondence network table RNA which corresponds from the network table NA of drawing 4 . And it asks for the correspondence terminal table RP 1 from the correspondence network table RNA of drawing 7 , and the terminal table corresponding to the correspondence terminal table RP 1 of drawing 6 is changed into the terminal table P16 of the input terminal A1 of component A10F mapped this time from the terminal table P1 of the input terminal I0 of component A10A.

[0056] Next, the terminal table of the supply place of the network table NA is changed into P16 from P1 at step S14 of drawing 10 . Furthermore, the network table which the terminal table P16 connects is set to NA at step S15.

[0057] Drawing 11 is Above MAPP (A10A.I0, A10F.A1). It is drawing showing the contents of the correspondence terminal table changed by processing, and the network table. The terminal table on which the correspondence terminal table RP 1 corresponds is changed into the terminal table P16 of component A10F mapped from P1 this time by this drawing, and the terminal table of the head of the supply place of the network table NA is also changed into P16 from P1 in it so that it may be shown.

[0058] By these processings, component A10A is permuted by component A10F, and it is saved that the input terminal A1 of component A10F of the compounded circuit diagram is equivalent to the input terminal I0 of component A10A of a functional diagram.

[0059] It is MAPP (A10A.I0, A10F.A1) as mentioned above. If processing is completed Next,

MAPP of step S3 of drawing 8 (A10A.I1, A10F.A2) MAPP of processing and step S4 (A10A.I2, A10F.A3) Processing and MAPP of step S5 (A10A.I2, A10F.A3) Processing is performed.

[0060] These MAPP processings are MAPP (A10A.I0, A10F.A1) mentioned above. It is processing and the same processing and is the processing which changes the input terminal I2 of component A10A into input terminal A3 of component A10F, and changes the output terminal A of component A10A into the input terminal A2 of component A10F which mapped the input terminal I0 of component AI0A this time at the output terminal X of component A10F.

[0061] If the input/output terminal of component A10A is changed into the input/output terminal of component A10F, the component table and each terminal table of component A10A will be deleted at the following step S6. Here, the store method of the correspondence relation between the component of the functional diagram when mapping AND macro A10A in AND cel A10F and the component of a circuit diagram is explained with reference to drawing 1 .

[0062] if mapping is performed, A10Ffrom A10A.A.X will change into this drawing (2), respectively. the live data corresponding to [so that it may be shown] each terminal of associated data A10A -- A10A.I0 from -- A10F.A1 A10A.I1 from -- A10F.A2 A10A.I2 from -- A10F.A3 Thereby, it is saved that input terminals A1 and A2, and A3 and the output terminal X of component A10F of a circuit diagram correspond to the input terminals I0, I1, and I2 and output terminal A of component A10A of a functional diagram.

[0063] Moreover, since there is no modification of a network in this case at mapping of 1 to 1, network data are this drawing (1). It is saved as [condition]. Next, drawing 12 (1) It is this drawing (2) about shown AND macro A10B and A10C, OR macro A10D, and inverter macro A10E. The actuation in the case of mapping in cel A10G, inverter cel A10H, and A10I which consist of two shown AND and NOR is explained with reference to the flow chart of drawing 13 .

[0064] At step S21 of drawing 13 , the component table of cel A10G mapped this time and the terminal table of I/O are created. Next, it is MAPP (A1alumnus.I0, A10G.A1) at step S22. Processing is performed and the input terminal I0 of AND macro A10B is changed into the input terminal A1 of cel A10G. Moreover, it is MAPP (A1alumnus.I1, A10G.B1) at step S23.

Processing is performed and the input terminal I1 of AND macro A10B is changed into the input terminal B1 of cel A10G.

[0065] It is MAPP (A1OC.I0, A10G.A2) at step S24 similarly. Processing is performed, the input terminal I0 of AND macro A10C is changed into the input terminal A2 of cel A10G, and it is MAPP (A1OC.I1, A10G.B-2) at step S25. Processing is performed and the input terminal I1 of AND macro A10C is changed into input terminal B-2 of cel A10G.

[0066] At the following step S26, the component table and the terminal table of an input/output terminal of inverter cel A10H are generated. And INSINV of step S27 (A10G.X, A10H.I, A10D.A, and A10H.X) Processing is performed.

[0067] Here, the above-mentioned INSINV processing is explained with reference to the flow chart of drawing 14 . First, the network table NL is created at step S31 of drawing 14 . Next, at step S32, the output terminal X of cel A10G is set up as a connection terminal of the supply origin of the created network table NL, and the input terminal I of inverter cel A10H is set up as a connection terminal of the supply place of the network table NL at step S33. It is registered as a network whose network nl connects the output terminal X which is cel A10G of a circuit diagram, and the input terminal I of inverter cel A10H by these processings.

[0068] Furthermore, the output terminal X of cel A10H is set up as a terminal of the supply origin of the network table NJ at the following step S34. Thereby, the supply origin of Network nj is changed into the output terminal X of component A1IH from the output terminal A of component A10D.

[0069] Moreover, the network nj of a functional diagram is drawing 12 (3). Since it is equivalent to what carried out logic reversal of the network nl of a circuit diagram, the network table NL newly created at step S35 as a network table which corresponds by logic reversal of the correspondence network table RNJ is set up.

[0070] At the following step S36, it is the correspondence terminal table (RP13) of the output terminal A of component A10D. About the pointer which points out a corresponding terminal table, it is the terminal table (P26) of the output terminal X of cel A10H. It changes so that it may point out. It is saved that the output terminal A of component A10D of a functional diagram and the output terminal X of component A10H of a circuit diagram correspond by this.

[0071] Moreover, since what carried out logic reversal of the output X of cel A10G newly

mapped in this case is equivalent to the output A of component A10D of a functional diagram, it is the following step S37, and it is the correspondence terminal table (RP13) of the output terminal A of component A10D of a functional diagram. As a terminal table which corresponds by logic reversal, it is the terminal table (P24) of the output terminal X of AI0G. It sets up.

[0072] Since what similarly carried out logic reversal of the input I of newly mapped in butter cel A10H is equivalent to the output A of component A10D of a functional diagram, it is step S38, and it is the correspondence terminal table (RP13) of the output terminal A of A10D. As a terminal table which corresponds by logic reversal, it is the terminal table (P25) of the input terminal I of A10H. It adds.

[0073] In addition, although not illustrated, in P24, P25 shows the correspondence terminal table of the input terminal I of inverter cel A10H, and P26 shows the correspondence terminal table of the output terminal X of inverter cel A10H for the correspondence terminal table of the output terminal X of cel A10G.

[0074] It is MAPP (A10E.I, A10I.I) at return and step S28 to drawing 13 . Processing is performed and the input terminal I of inverter macro A10E is changed into the input terminal I of inverter cel A10I.

[0075] At step S29, it is MAPP (A10E.X, A10I.X) similarly. Processing is performed and the output terminal X of inverter macro A10E is changed into the output terminal X of inverter cel A10I.

[0076] Since the networks nh and ni of a circuit diagram, and component A10B, A10C, A10D and A10E stopped existing by having performed mapping of cel A10G, inverter cel A10H, and A10I, the network tables NH and NI of live data, and component A10B, A10C, A10D and the component table and each terminal table of A10E are deleted at step S30.

[0077] Here, the store method of the correspondence relation between the functional diagram at the time of many to many mapping mentioned above and a circuit diagram is explained with reference to drawing 12 . When mapping of cel A10G, A10H, and A10I is performed, it is A10B.I0 of associated data, and A10B.I1. It is A10G.A1 and A10G.B1 as corresponding live data. It is saved. A10C.I0 of associated data, and A10C.I1 They are A10G.A2 and A10G.B-2 as corresponding live data. It is saved.

[0078] Thereby, the input terminals I0 and I1 of component A10B of a functional diagram,

the input terminals I0 and I1 of component A10C, the input terminals A1, B1, and A2 of component A10G of a circuit diagram, and correspondence relation with B-2 are saved. Therefore, when performing the logic simulation of a circuit diagram, it becomes possible to specify the input terminals I0 and I1 of component A10B of a functional diagram, and the input terminals I0 and I1 of component A10C as the trace point.

[0079] Moreover, since A10G.X and A10H.I are saved with the information which shows that logic is reversed as live data corresponding to A10D.A of associated data, those information shows that the output of component A10D of a functional diagram is equivalent to what carried out logic reversal of the output X of component A10G of a circuit diagram, and the thing which carried out logic reversal of the input I of component A10H.

[0080] Furthermore, although the terminal corresponding to the output terminal A of component A10B of a functional diagram, the output terminal A of component A10C, and the input terminals I0 and I1 of component A10D stops existing on a circuit diagram in this case, those terminal information on the component of a functional diagram (associated data) is saved as they are.

[0081] Next, in order to optimize a circuit, drawing 15 explains the actuation in the case of deleting two inverter cels connected to the serial with reference to the flow chart of drawing 16 . At step S41 of drawing 16 , it asks for the correspondence network table RNK from the network table NK of live data. And at the following step S42, what points out the terminal table of the output terminal X of inverter cel A10I which it is going to delete this time in the pointer of the correspondence terminal table linked to the correspondence network table RNK is looked for, and the pointer is changed so that the terminal table of the output terminal X of component A10G may be pointed out.

[0082] Thereby, the terminal of the circuit diagram corresponding to the output terminal X of component A10E linked to the network nk of a functional diagram is changed into the output terminal X of component A10G from the output terminal X of inverter cel A10I.

[0083] Next, at step S43, it asks for the correspondence network table RNJ from the network table NJ of live data, and in the pointer of the correspondence terminal table linked to the correspondence network table RNJ, what points out the terminal table of the input terminal I of inverter cel A10I which it is going to delete this time is looked for, and the pointer and terminal table of an input terminal I are deleted by step S44. Furthermore, the output terminal

X of component A10G is added as a terminal table connected by logic reversal of the correspondence terminal table.

[0084] It is changed so that the output terminal X of component A10G of a circuit diagram may correspond to the input terminal I of component A10E of the functional diagram linked to Network nj by logic reversal by this. Next, at step S45, in the pointer of the correspondence terminal table linked to the correspondence network table RNJ, what points out the output terminal X of inverter A10H which it is going to delete this time is looked for, and the pointer and the terminal table of the output terminal X of inverter A10H are deleted.

[0085] Thereby, from the terminal table of the output terminal A of component A10D of the functional diagram linked to Network nj, the pointer which points out the output terminal X of inverter A10H of a circuit diagram, and the pointer which points out the input terminal I of inverter A10H which corresponds by logic reversal are deleted, and correspondence relation with the input terminal I of inverter A10H which corresponds by the output terminal A of component A10D, the output terminal X of inverter A10H, and logic reversal is canceled.

[0086] In addition, the terminal table of the output terminal X of component A10G is set to the terminal table of the output terminal A of component A10D of a functional diagram by processing (mapping processing of component A10G grade) mentioned above as a terminal table of the circuit diagram which corresponds by logic reversal, and it is set up that the output terminal X of component A10G of a circuit diagram corresponds to the output terminal A of component A10D of a functional diagram by logic reversal.

[0087] Next, at step S46, in the pointer of the correspondence terminal table linked to the correspondence network table RNJ, what points out the terminal table of the input terminal I of inverter A10H which it is going to delete this time is looked for, and the pointer and the terminal table of the input terminal I of component A10H are deleted.

[0088] Thereby, from the correspondence terminal table of the input terminal I of component A10D of the functional diagram linked to Network nj, the pointer which points out the input terminal I of inverter A10H is deleted, and the correspondence relation between the input terminal I of component A10D of a functional diagram and the input terminal I of inverter A10H of a circuit diagram is canceled.

[0089] Next, at step S47, network table NK deletion is carried out as a network table on which the correspondence network table RNK corresponds, and the network table NL is set up

instead. Next, NJ is deleted from the pointer which points out the network table on which the correspondence network table RNJ corresponds at step S48, and the network table NL is added as a network table which corresponds by logic reversal.

[0090] Since modification of connection of Networks nk and nj was completed by this, the network tables NK and NJ are deleted at step S49. Furthermore, the component table of A10H and the component table of A10I are deleted at step S50.

[0091] Here, the correspondence-related store method of the functional diagram and circuit diagram when deleting inverter cel A10H and A10I for optimization of a circuit is explained with reference to drawing 15 . In this case, since component A10H of a circuit diagram and A10I are deleted, live-data A10H.I corresponding to that output terminal A is deleted by logic reversal with live-data A10H.X corresponding to the live data A of those components, i.e., the output terminal of component A10D of a functional diagram, (associated data A10D.A).

[0092] Furthermore, since the input terminal I of component A10E of a functional diagram is equivalent to what reversed the logic of the output terminal X of component A10G of a circuit diagram, live-data A10I.I (drawing 15 (3)) corresponding to associated data A10E.I is deleted, and it is saved that live-data A10G.X corresponds by logic reversal instead (this drawing (4)).

[0093] Moreover, since the output terminal X of component A10E is equivalent to the output terminal of component A10G, live-data A10I.I (this drawing (3)) corresponding to associated data A10E.I is deleted, and live-data A10G.X is saved instead (this drawing (4)).

[0094] By thus, the thing for which correspondence relation with the component of the circuit diagram corresponding to the terminal and terminal of the component of a functional diagram is saved by logic reversal Even when the component of a circuit diagram is deleted for optimization of a circuit and the component corresponding to the component of a functional diagram stops existing in a circuit diagram, it becomes possible to specify the terminal of the component of a functional diagram as the trace point by saving correspondence relation with the terminal of the circuit diagram which corresponds by logic reversal.

[0095] Next, the actuation in the case of performing simulation based on the functional diagram and circuit diagram which were created by the correspondence relation store method mentioned above is explained with reference to drawing 17 and drawing 18 . Drawing 17 is the explanatory view of the simulation of a functional diagram, First, the trace point which

carries out simulation is specified. This drawing shows the case where the input terminals I0, I1, and I2, the output terminal X, and Network nj of an AND macro of drawing 1 are specified as the trace point. [of a functional diagram]

[0096] Next, if the test pattern for simulation is chosen, simulation will be performed by the simulator of a functional diagram and a simulation result as shown in drawing 17 ** will be displayed.

[0097] In performing simulation of gate level, as shown in drawing 18 **, it specifies the terminal and network on a functional diagram as the trace point. Then, the terminal and network of a circuit diagram corresponding to the trace point specified on the functional diagram are called for from the data in which the correspondence relation of a functional diagram and a circuit diagram as shown in drawing 15 is shown by the conversion program.

[0098] Since the data in which the correspondence relation of the terminal of drawing 15 is shown show that the input terminals A1 and A2, A3, and the output terminal X of component A10F of a circuit diagram deal with the input terminals I0, I1, and I2 and output terminal X of component A10A of a functional diagram, those terminals on a circuit diagram are specified as the trace point. Furthermore, since the data in which the correspondence relation of the network of drawing 15 is shown show that the network nj of a functional diagram is equivalent to the network nl of a circuit diagram by logic reversal, it indicates that logic with Networks nj and nl is reversed by the message.

[0099] And by the simulator of gate level, a predetermined simulation test pattern is performed and the simulation result of a functional diagram and the simulation result of gate level are displayed. By comparing both simulation result, it can check whether logic of a functional diagram is correctly realized also on gate level.

[0100] According to the correspondence relation store method of a functional diagram and a circuit diagram mentioned above, since the input/output terminal of the input/output terminal of the component of a functional diagram and the component of a network and a circuit diagram and correspondence relation with a network can be saved, the terminal and network of a functional diagram can be specified as the trace point of the simulation of gate level (component of a circuit diagram), and it can investigate easily whether the logic of a functional diagram and the circuit of gate level is in agreement.

[0101] In addition, the approach of saving the correspondence relation between a functional

diagram and a circuit diagram may be realized not only with the configuration of the table of the example mentioned above but with other configurations.

[0102]

[Effect of the Invention] According to this invention, the correspondence relation between a functional diagram and a circuit diagram can be more correctly saved by saving [include], also when logic reversal produces the correspondence relation between the terminal of the component of a functional diagram and a network, and the terminal of the component of a circuit diagram and a network between a terminal or a network. Thereby, when performing simulation of a circuit diagram, the input/output terminal and network of a component of a functional diagram can be specified as the simulation point, and the check and debugging of the circuit after mapping can be performed efficiently.

[Translation done.]

CLAIMS

[Claim(s)]

[Claim 1] The correspondence relation store method of the functional diagram and the circuit diagram which carry out [saving the correspondence relation between the input/output terminal of the component of said functional diagram, and the input/output terminal of the component of said circuit diagram with logic reversal information, and saving the correspondence relation between the network of said functional diagram, and the network of said circuit diagram with logic reversal information in the approach of saving the correspondence relation between the component independent of technology, the functional diagram constituted by the network, the component depending on technology, and the circuit diagram which are constituted by the network, and] as the description.

15

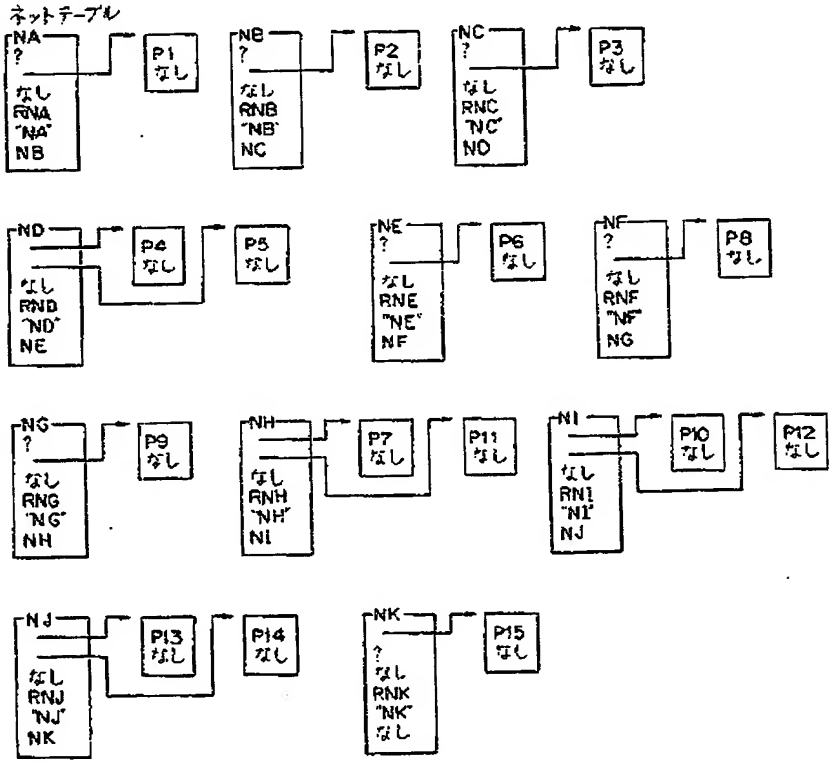
（4）である。
【図15】機能図及び回路図と保存されるデータを示す図（3）である。
【図16】実施例の対応関係保存方法のフローチャート（5）である。
【図17】機能図のシミュレーションの説明図である。
【図18】ゲートレベルのシミュレーションの説明図である。
【図19】従来の対応関係の保存方法の説明図（1）である。

16

＊【図20】従来の対応関係の保存方法の説明図（2）である。
【符号の説明】
E1、E2　素子テーブル
P1、P2　端子テーブル
NA、NB　ネットテーブル
RA10A、RA10B　対応素子テーブル
RP1、RP2　対応端子テーブル
RNA、RNB　対応ネットテーブル

【図4】

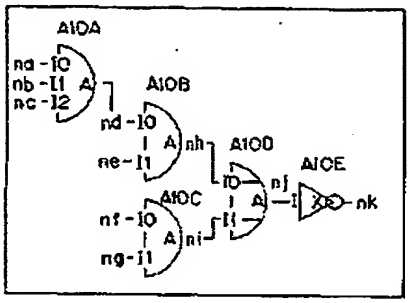
ネットテーブルの具体的内容を示す図



【図1】

機能図及び回路図と保存されるデータを示す図(1)

(1)

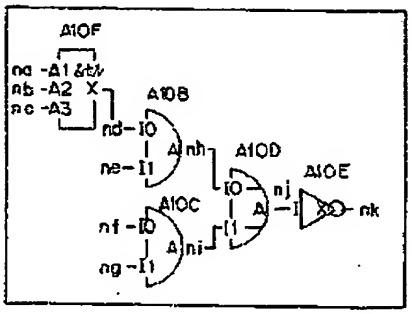


<素子> 対応データ	実データ	<ネット> 対応データ	実データ
A10A.I0	A10A.I0	na	na
A10A.I1	A10A.I1	nb	nb
A10A.I2	A10A.I2	nc	nc
A10A.A	A10A.A	nd	nd
A10B.I0	A10B.I0	ne	ne
A10B.I1	A10B.I1	nf	nf
A10B.A	A10B.A	ng	ng
A10C.I0	A10C.I0	nh	nh
A10C.I1	A10C.I1	ni	ni
A10C.A	A10C.A	nj	nj
A10D.I0	A10D.I0	nk	nk
A10D.I1	A10D.I1		
A10D.A	A10D.A		
A10E.I	A10E.I		
A10E.X	A10E.X		

対応データの端子から、実データの対応する端子をさす。

↓ A10Aをマッピングする。
A10AのANDマクロはA10FのANDセルに1対1でマッピングされる。
マッピング時に、対応データのA10AのI0に対応する実データの端子をA10FのA1に変更する。
I1,I2,Aについても同様。

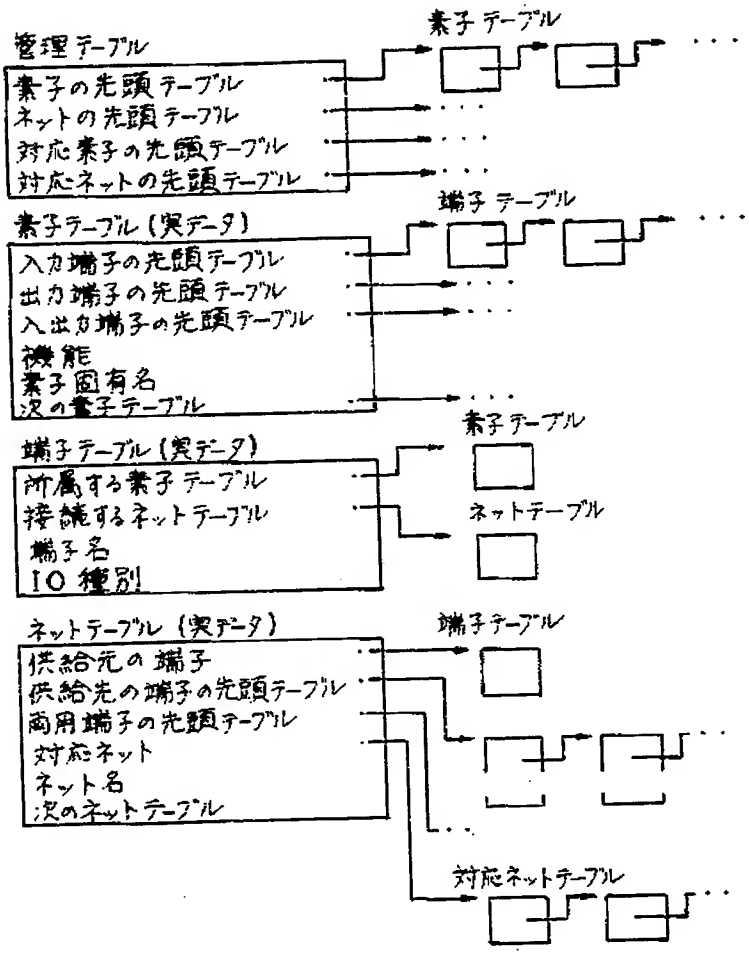
(2)



<素子> 対応データ	実データ	<ネット> 対応データ	実データ
A10A.I0	A10F.A1	na	na
A10A.I1	A10F.A2	nb	nb
A10A.I2	A10F.A3	nc	nc
A10A.A	A10F.X	nd	nd
A10B.I0	A10B.I0	ne	ne
A10B.I1	A10B.I1	nf	nf
A10B.A	A10B.A	ng	ng
A10C.I0	A10C.I0	nh	nh
A10C.I1	A10C.I1	ni	ni
A10C.A	A10C.A	nj	nj
A10D.I0	A10D.I0	nk	nk
A10D.I1	A10D.I1		
A10D.A	A10D.A		
A10E.I	A10E.I		
A10E.X	A10E.X		

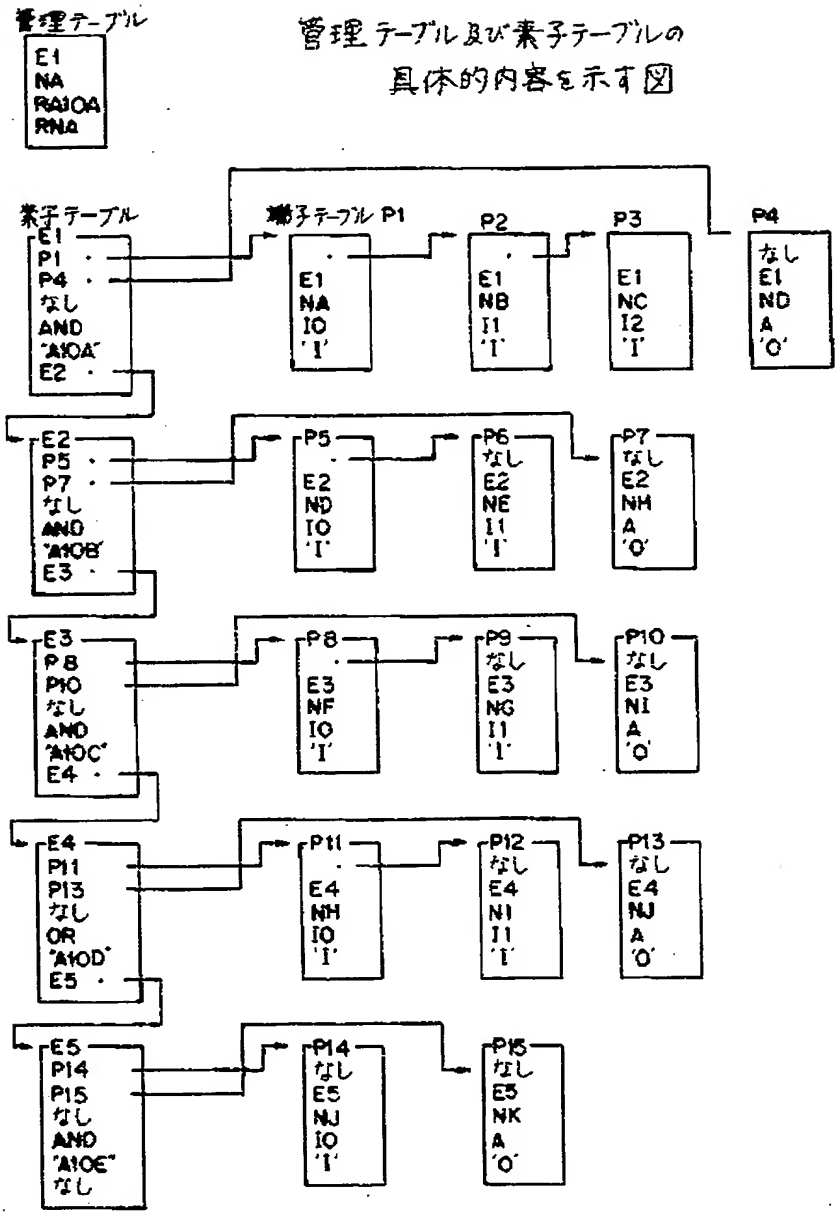
【図2】

テーブルの構成を示す図(1)



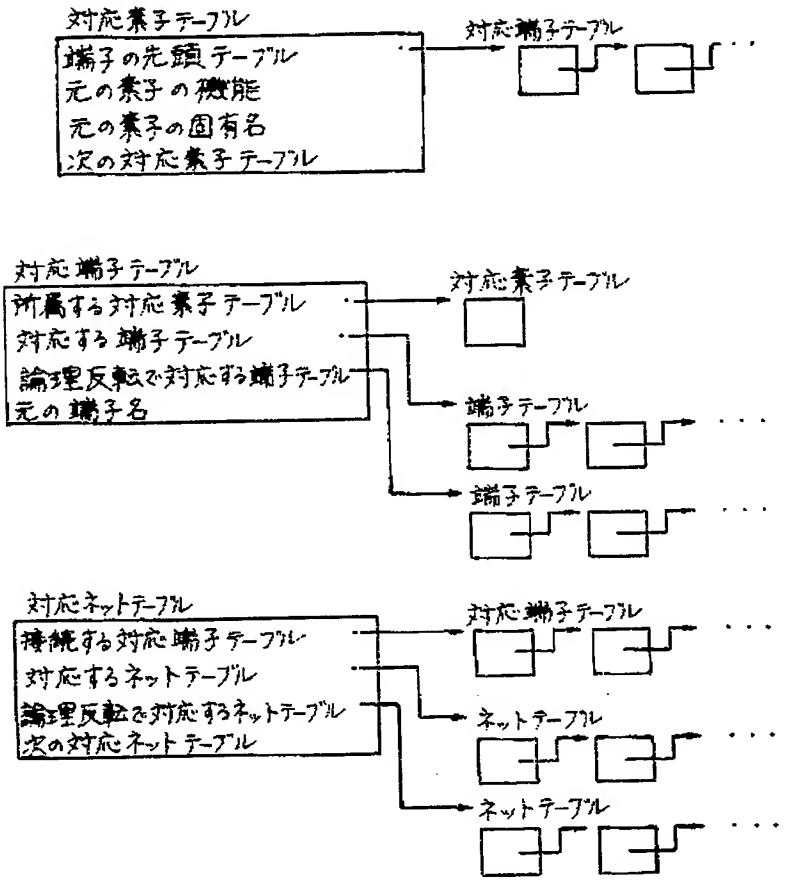
【図3】

管理テーブル及び素子テーブルの具体的内容を示す図



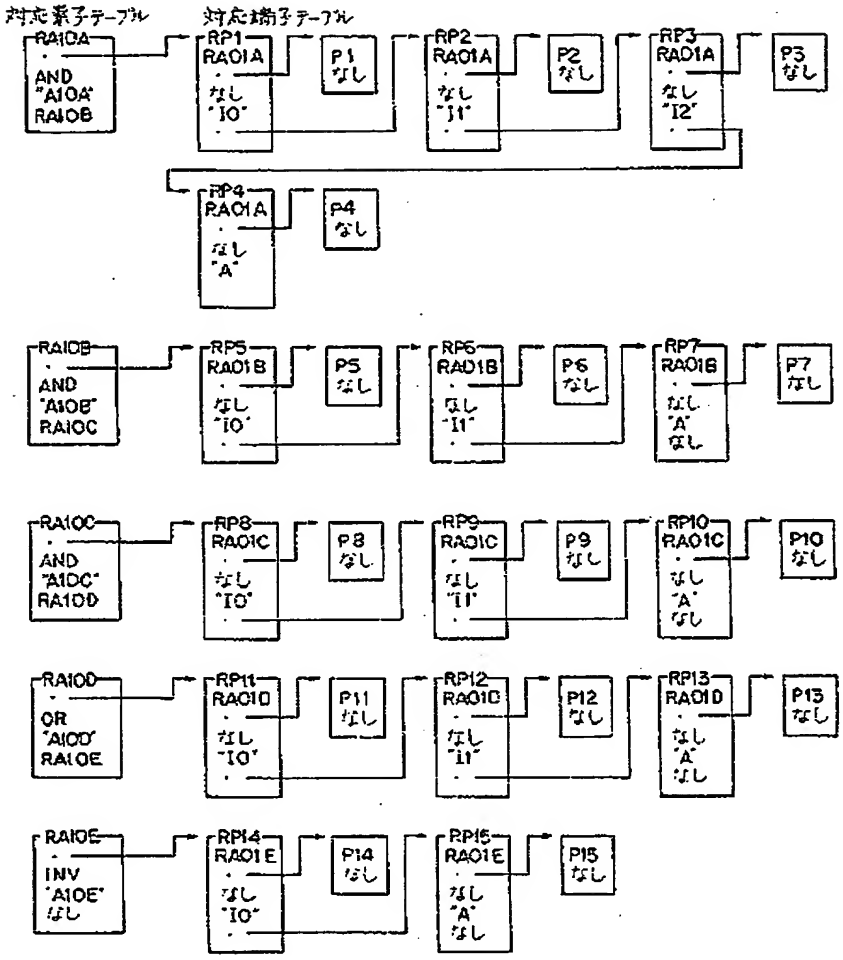
【図5】

テーブルの構成を示す図(2)



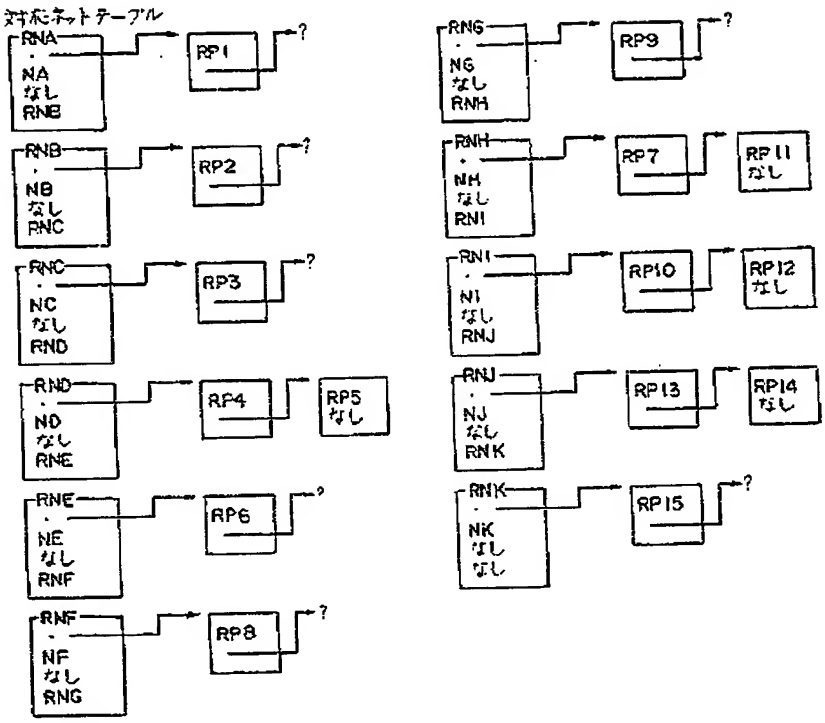
【図6】

対応素子テーブルの具体的内容を示す図



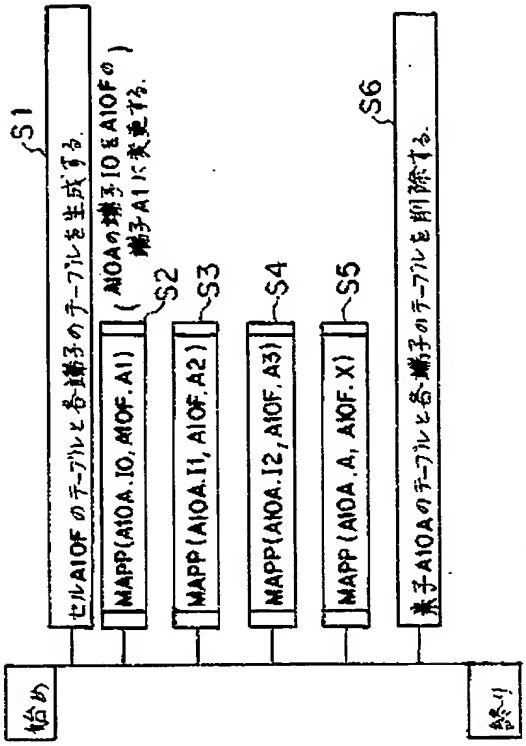
【図7】

対応ネットテーブルの具体的内容を示す図



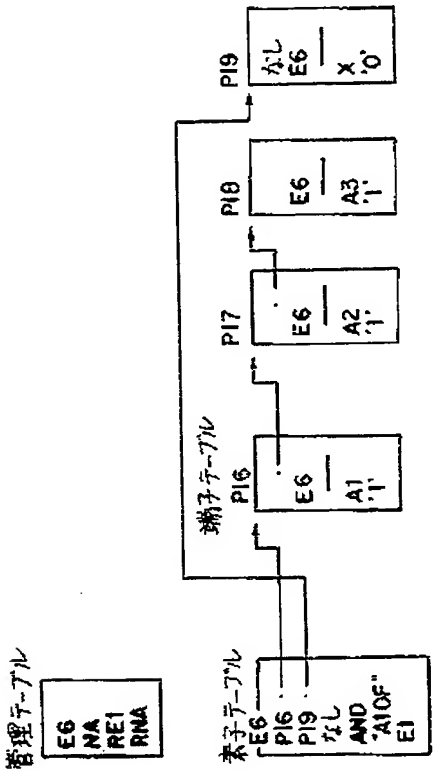
【図8】

実施例の対応関係保存方法の
フローチャート(1)



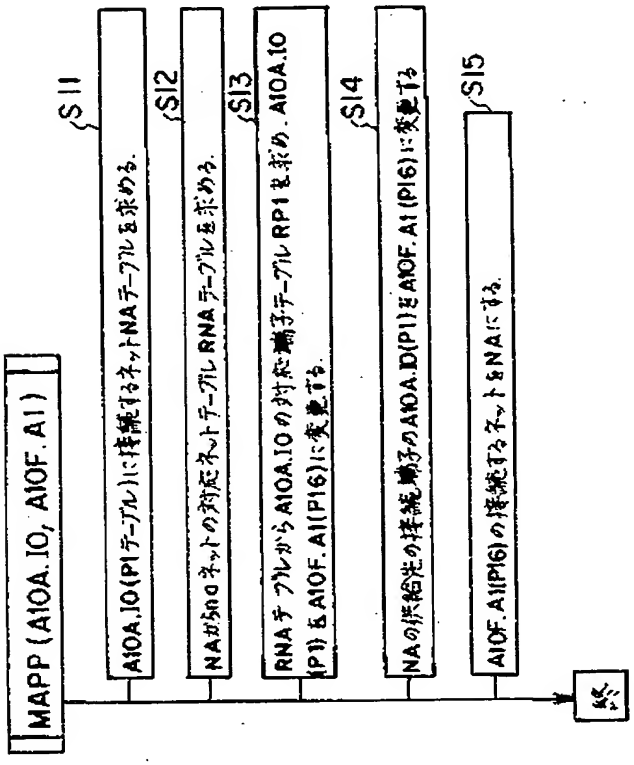
【図9】

ANDセルA1OFのマッピング時に作成される
素子テーブルと管理テーブルの内容を示す図



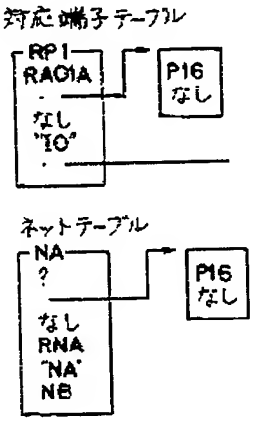
【図10】

実施例の対応関係保存方法のフローチャート(2)



【図11】

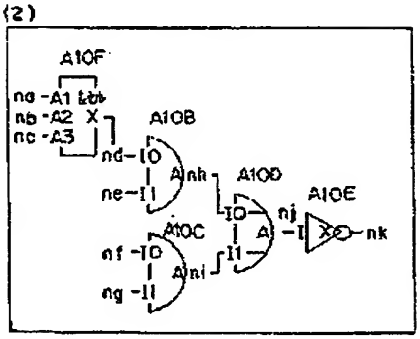
MAPP(A10A,10,A10F,A1)処理によりリンクが変更された対応端子テーブルとネットテーブルの内容を示す図



【図12】

機能図及び回路図と保存されるデータを示す図 (2)

(2)



<素子>

対応データ	実データ
A10A.I0	A10F.A1
A10A.I1	A10F.A2
A10A.I2	A10F.A3
A10A.A	A10F.X
A10B.I0	A10B.I0
A10B.I1	A10B.I1
A10B.A	A10B.A
A10C.I0	A10C.I0
A10C.I1	A10C.I1
A10C.A	A10C.A
A10D.I0	A10D.I0
A10D.I1	A10D.I1
A10D.A	A10D.A
A10E.I	A10E.I
A10E.X	A10E.X

<ネット>

対応データ	実データ
na	na
nb	nb
nc	nc
nd	nd
ne	ne
nf	nf
ng	ng
nh	nh
ni	ni
nj	nj
nk	nk

↓ ①A10B, A10C, A10Dを A10G, A10Hにマッピングする.

A10Bの10の対応	を A10GのA1
I1	B1
A	なし
A10C I0	A2
I1	B2
A	なし
A10D I0	なし
I1	なし
A	A10HのX (反転論理 A10GのX. A10HのI)

に変更する.

ネットniはネットnjと対応づける. ただし論理反転.

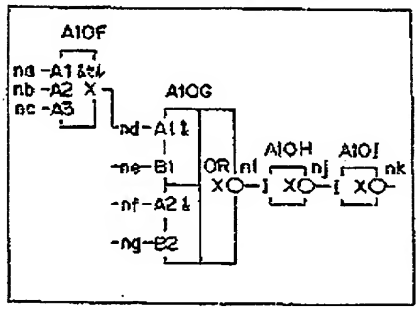
② A10Eを A10Iにマッピングする

A10EのIの対応を A10IのI

A10EのXの対応を A10IのX

に変更する.

(3)



<素子>

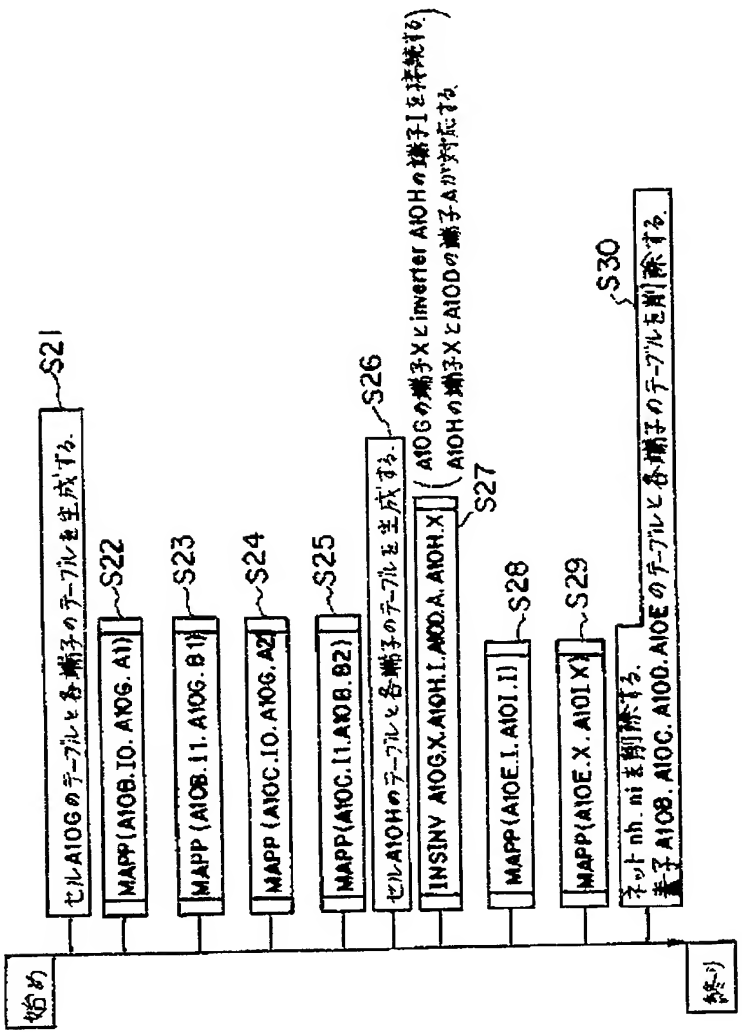
対応データ	実データ
A10A.I0	A10F.A1
A10A.I1	A10F.A2
A10A.I2	A10F.A3
A10A.A	A10F.X
A10B.I0	A10G.A1
A10B.I1	A10G.B1
A10B.A	
A10C.I0	A10H.A2
A10C.I1	A10H.B2
A10C.A	
A10D.I0	
A10D.I1	
A10D.A	A10H.X
A10D.A (反転)	A10G.X
A10D.A (反転)	A10H.I
A10E.I	A10I.I
A10E.X	A10I.X

<ネット>

対応データ	実データ
na	na
nb	nb
nc	nc
nd	nd
ne	ne
nf	nf
ng	ng
nh	—
ni	—
nj	nj
nk	nk
nj (反転)	ni

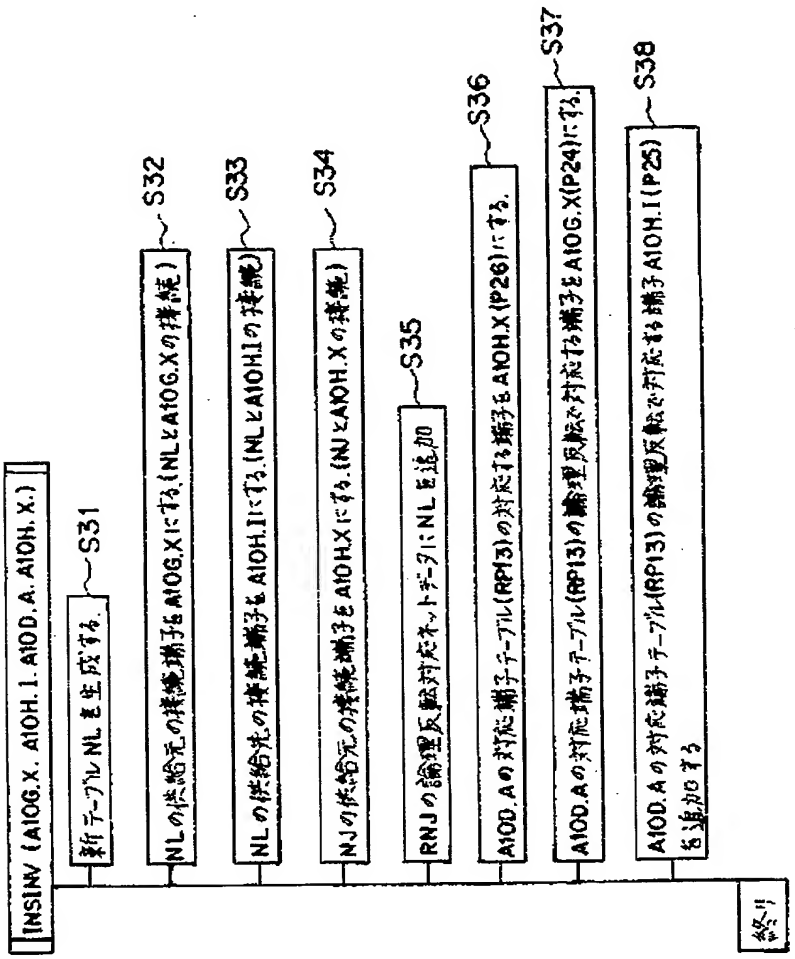
【図13】

実施例の対応関係保存方法の
フローチャート(3)



【図14】

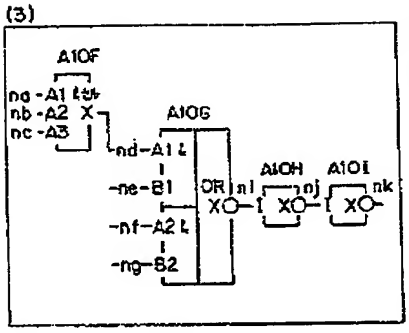
実施例の対応関係保存方法のフローチャート(4)



【図15】

機能図及び回路図と保存されるデータを示す図 (3)

(3)



<素子>

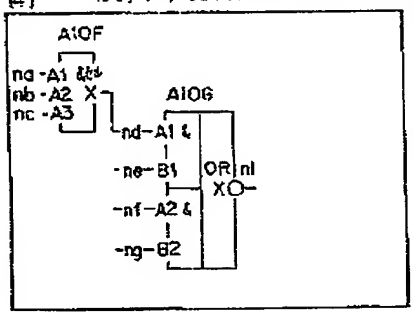
対応データ	実データ
A10A.I0	A10F.A1
A10A.I1	A10F.A2
A10A.I2	A10F.A3
A10A.A	A10F.X
A10B.I0	A10G.A1
A10B.I1	A10G.B1
A10B.A	
A10C.I0	A10H.A2
A10C.I1	A10H.B2
A10C.A	
A10D.I0	
A10D.I1	
A10D.A	A10H.X
A10D.A(反転)	A10G.X
A10D.A(反転)	A10H.I
A10E.I	A10I.I
A10E.X	A10I.X

<ネット>

対応データ	実データ
na	na
nb	nb
nc	nc
nd	nd
ne	ne
nf	nf
ng	ng
nh	—
ni	—
nj	nj
nk	nk
nj (反転)	ni

↓ inv-inv の余分な inv を圧縮する.
A10DのAの対応データのA10HのX, A10H.Iを削除する
そのとき, A10EのXの対応をA10GのXに変更し, A10EのXの対応もA10GのX
(反転)に変更する. A10H, A10Iが消えても, A10D, A10Eの対応データは消えない.

(4)



<素子>

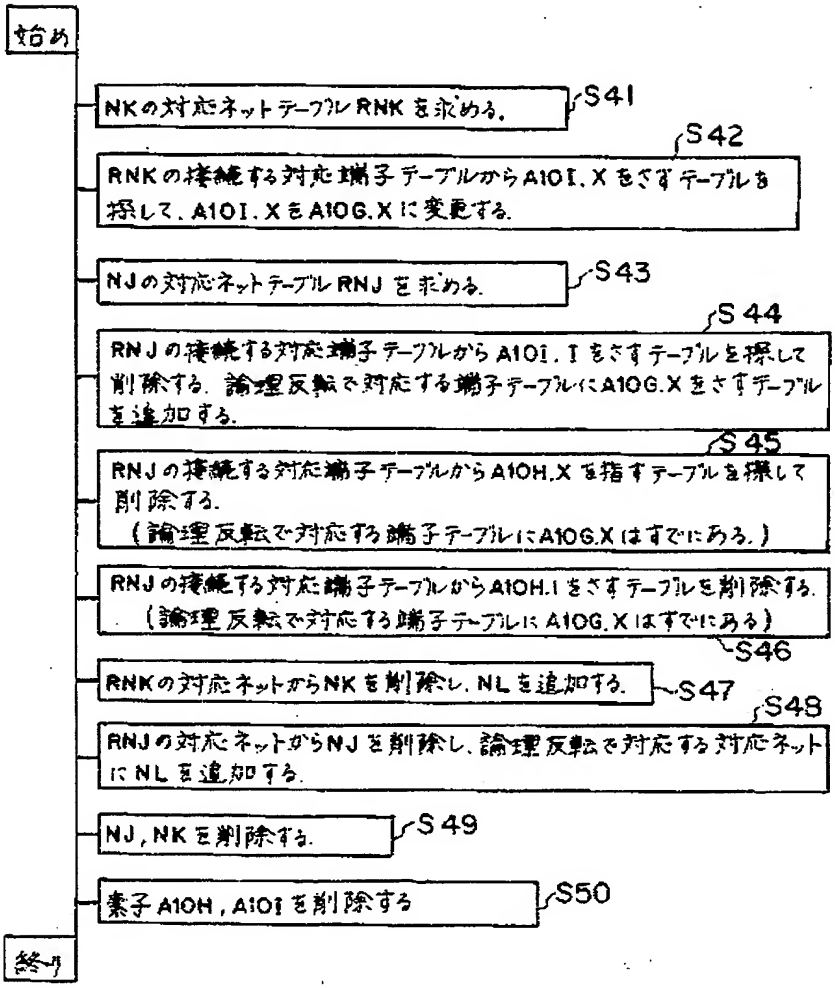
対応データ	実データ
A10A.I0	A10F.A1
A10A.I1	A10F.A2
A10A.I2	A10F.A3
A10A.A	A10F.X
A10B.I0	A10G.A1
A10B.I1	A10G.B1
A10B.A	
A10C.I0	A10G.A2
A10C.I1	A10G.B2
A10C.A	
A10D.I0	
A10D.I1	
A10D.A	
A10D.A(反転)	A10G.X
A10E.I(反転)	A10G.X
A10E.X	A10G.X

<ネット>

対応データ	実データ
na	na
nb	nb
nc	nc
nd	nd
ne	ne
nf	nf
ng	ng
nh	—
ni	—
nj (反転)	ni
nk	ni

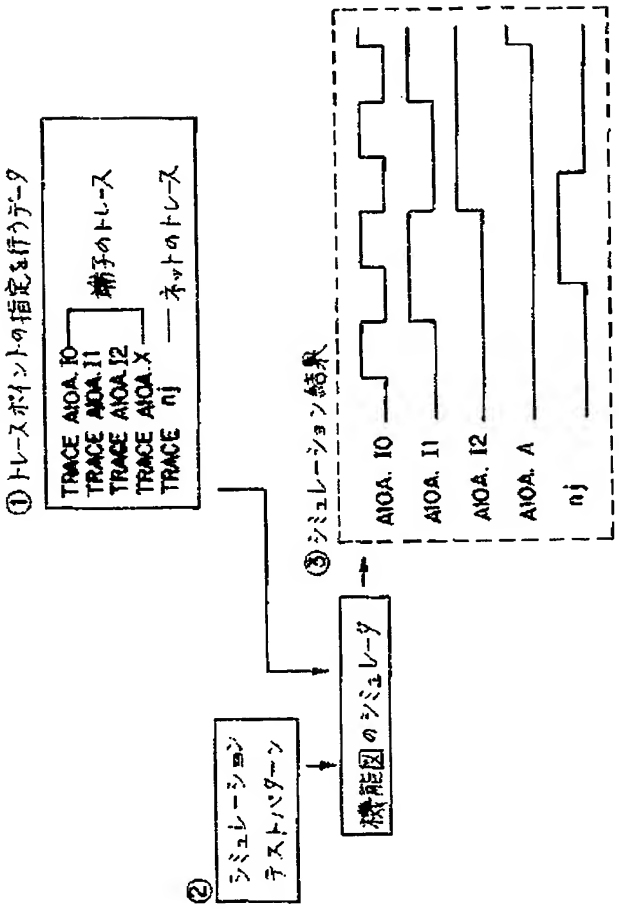
【図16】

実施例の対応関係保存方法のフローチャート(5)



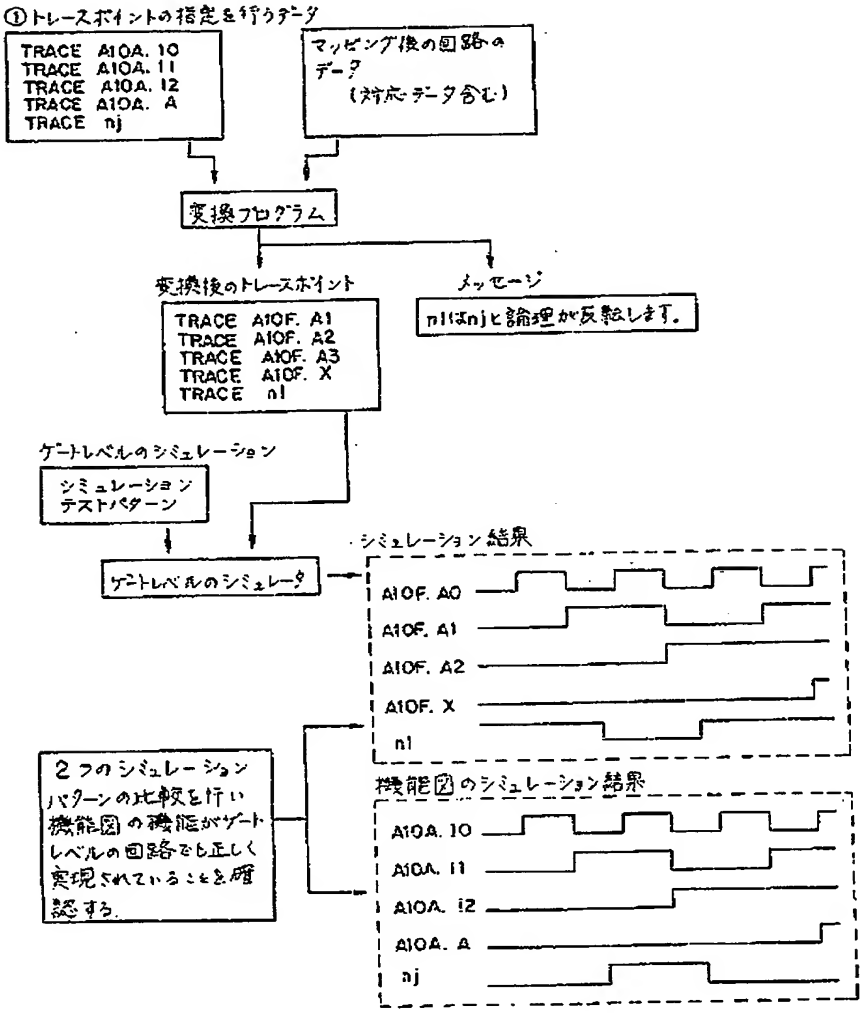
【図17】

機能図のシミュレーションの説明図



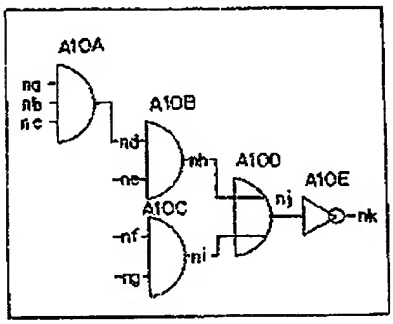
【図18】

ゲートレベルのシミュレーションの説明図



【図19】

従来の対応関係の保存方法の説明図 (1)



<素子>

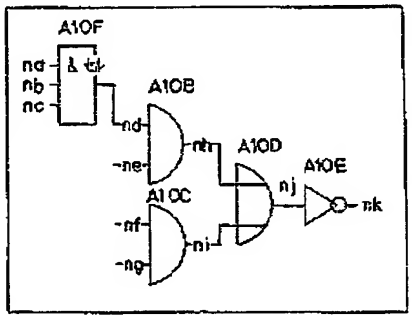
実データ	対応データ
A10A	A10A
A10B	A10B
A10C	A10C
A10D	A10D
A10E	A10E

<ネット>

実データ	対応データ
na	na
nb	nb
nc	nc
nd	nd
ne	ne
nf	nf
ng	ng
nh	nh
ni	ni
nj	nj
nk	nk

A10Aをマッピングする
A10AのANDマクロはA10FのANDセル1対1でマッピングされる.
A10AからA10Fに置き換えるとき,対応する素子のデータをA10AからA10Fに変更する.

(2)



<素子>

実データ	対応データ
A10F	A10A
A10B	A10B
A10C	A10C
A10D	A10D
A10E	A10E

<ネット>

実データ	対応データ
na	na
nb	nb
nc	nc
nd	nd
ne	ne
nf	nf
ng	ng
nh	nh
ni	ni
nj	nj
nk	nk

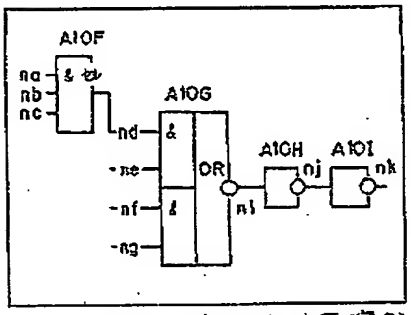
【図20】

従来の対応関係の保存方法の説明図(2)

① A10B, A10C, A10D を A10G, A10H にマッピングする
対応する素子は1つしか指せないので、A10G, A10H, の対応データは A10D となる。
機能図 A10B, A10C と対応する実データは無くなる。
ネット nh と ni は無くなり、ni に対応するネットは対応データには無い

② A10E を A10I にマッピングする。
A10E から A10I に置き換えるとき、対応する素子のデータを A10E から A10I に変更する

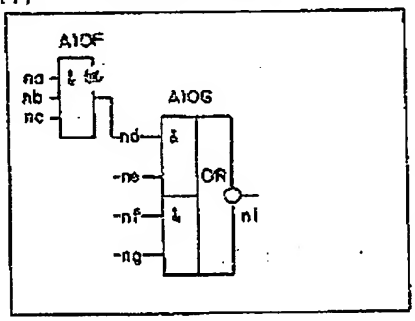
(3)



<素子>		<ネット>	
実データ	対応データ	実データ	対応データ
A10F	A10A	na	na
—	A10B	nb	nb
—	A10C	nc	nc
A10G	A10D	nd	nd
A10H	A10D	ne	ne
A10I	A10E	nf	nf
		ng	ng
		—	nh
		—	ni
		nj	nj
		nk	nk
		ni	—

inv–inv の余分な inv を圧縮する。
ネット nj, nk に対応するネットが無くなる。

(4)



<素子>		<ネット>	
実データ	対応データ	実データ	対応データ
A10F	A10A	na	na
—	A10B	nb	nb
—	A10C	nc	nc
A10G	A10D	nd	nd
		ne	ne
		nf	nf
		ng	ng
		—	nh
		—	ni
		—	nj
		—	nk
		ni	—